Organize your work with viki and bibtex

Tom Link

08. Nov 2009

Contents

1 This Tutorial

When doing research, you usually collect papers, PDF or HTML documents and you comment on these papers or summarize them so that it is easier for you to find the relevant citation later on. In this tutorial, we use VIM and a few plugins (viki, tbibtools etc.) to organize all those files, comments, and notes.

2 Prerequisites

Since this tutorial uses VIM and BibTeX, you should have some knowledge of the VIM editor and the BibTeX file format. You don't have to have BibTeX or LaTeX actually installed. Some plugins may require a version of VIM with ruby support compiled in.

For VIM, you also need the following plugins installed:

- viki
- tlib

Optional plugins (search http://vim.sf.net for alternatives if you don't like them):

- ttoc
- tbibtools (requires a +ruby enabled `VIM`)
- trag
- tskeleton

Since most wikis support "interwikis", it should be quite easy to adapt this approach for other wikis.

3 Use a BibTeX-File As Entry Point

A typical `BibTeX` files contains entries like the following[1]:

```
@article{solka08,
    author       = {Jeffrey L. Solka},
    title        = {Text Data Mining: Theory and Methods},
    journal      = {Statistics Surveys},
    year         = {2008},
    volume       = {2},
    pages        = {94-112},
    url          = {http://projecteuclid.org/euclid.ssu/1216238228 },
    keywords     = {text mining; statistics},
    abstract     = {This paper provides the reader ...},
}
```

In order to facilitate searching the `BibTeX` file, I suggest to edit the `BibTeX` file with `tw=0`.

It depends on the `BibTeX` style defined in the LaTeX source document, how the `BibTeX` entry is eventually formatted for inclusion in your LaTeX document. Fields your `BibTeX` style does not know about will be ignored by `BibTeX`. You can make use of this behaviour to include additional information in the `BibTeX` entry.

```
@article{solka08,
    author       = {Jeffrey L. Solka},
    title        = {Text Data Mining: Theory and Methods},
    journal      = {Statistics Surveys},
    year         = {2008},
```

[1] I use this paper as an example because I recently added it to my `BibTeX` file and because I'm too lazy to come up with an contrived entry.

```
    volume       = {2},
    pages        = {94-112},
    url          = {http://projecteuclid.org/euclid.ssu/1216238228 },
    keywords     = {text mining; statistics},
    abstract     = {This paper provides the reader ...},
    summary      = {[[SMRY::solka08]]},
    file         = {[[SDOCS::Statistics Surveys/2008/solka08.pdf]]},
    doi          = {[[DOI::10.1214/07-SS016]]},
}
```

The additional fields `summary`, `file`, and `doi` contain `viki` links that refer to `intervikis`. In order to turn those extra fields into hyperlinks, you can use the `:VikiMinorMode` in `VIM`. In order to always invoke this command when editing a `BibTeX` file, you could create a file `~/vimfiles/after/syntax/bib.vim` (where `vimfiles` refers to your personal `VIM` runtime, but see `:help vimfiles` in `VIM`):

```
VikiMinorMode
```

The intervikis are best defined in `~/vimfiles/after/plugin/viki.vim`:

```
call viki#Define('SMRY',   '~/Sci/Summaries', '.txt')
call viki#Define('SDOCS',  '~/Sci/Docs')
call viki#Define('DOI',    'http://dx.doi.org')
```

Move the cursor over the link to the summary and press <c-cr> or <a-leftmouse>. This will create or open the file `~/Sci/Summaries/solka08.txt` that will contain you notes on the (randomly chosen) paper with the `BibTeX` ID solka08.

4 Search `BibTeX` Files

Due to the format of the `BibTeX` file, the standard `VIM` tools for text search lead to suboptimal results. There are several tools around that facilitate searching `BibTeX` files. If you want a GUI, you could use, e.g., the java-based `JabRef` – be aware though that `JabRef` usually resorts your `BibTeX` files.

The `VIM` plugin `tbibtools`, provides a command `:TBibList` that creates a browsable list of the `BibTeX` entries including the entry's ID, its title, its authors or editors, as well as the entry's keywords. `tbibtools` includes a `ruby`-based command-line tool that can be used to sort or merge `BibTeX` files.

You can use `VIM` plugins like `ttoc` to get a quick overview of the entries contained in the `BibTeX` file opened in the current buffer.

5 Collect Summaries in a Directory

As a result of the interviki definitions, all your PDF files, summaries, and notes will be collected in well defined places. The advantage of collecting your notes as plain text files in a single directory is that you can use standard text tools like grep to search those notes.

VIM provides the :vimgrep command that could be used for that. The VIM plugin trag serves as a wrapper around vimgrep that could come handy.

I personally use tSkeleton to create new files with a template. A template for summaries could look like this:

```
<+CURSOR+>#AU: <+TEXT AUTHOR+>
#TI: <+TEXT TITLE+>
#DATE: <+YEAR+>
% #VAR: keywords=
% [[BIB::monos#<+FILE NAME ROOT:s/\..*$//+>]]

#s<+PAGE+>
```

A newly created file will then look like this:

```
#AU: <+TEXT AUTHOR+>
#TI: <+TEXT TITLE+>
#DATE: <+YEAR+>
% #VAR: keywords=
% [[BIB::monos#solka08]]

#s<+PAGE+>
```

The BIB interviki points to the directory where the BibTeX files are located. You can thus jump easily from the summary to the BibTeX file and back again. After having filled in the standard place holders, such a summary could look like this:

```
#AU: Jeffrey L. Solka
#TI: Text Data Mining: Theory and Methods
#DATE: 2008
#VAR: keywords=text mining; statistics
% [[BIB::monos#solka08]]

#s94
...
```

6 Generate HTML, PDF output

The viki markup is capable of handling citations and footnotes. A sample input file could look like this:

```
#AU: Tom Link
#TI: Whatever
#DATE: today
#BIB: mybib.bib

#Abstract <<---
This paper provides a demonstration of the deplate converter.
---

* Context
We all know $E = m \cdot c^2${cite: einstein}{fn: einstein}.

#Footnote id=einstein <<---
Or so we think.
---

#MAKEBIB: apa-style
```

You can use zM or :TToC to get a quick overview of the document structure.

The deplate converter can be used to convert your notes, summaries, and documents from viki markup to HTML or PDF (via LaTeX).

Change to the directory where the source files are:

```
cd ~/Sci/...
```

Initialize the directory (we add the particle-math module to enable embedding LaTeX-math syntax with a dollar sign):

```
deplate -m particle-math -m makefile FILENAME.txt
```

Generate HTML:

```
make
```

Generate PDF:

```
make pdf
```

Generate output for a file other than the one named when initializing the directory:

```
make FILE=OTHER_FILE.txt html
```